What is claimed is:

1. In a network processor for switching data between an input and an output, a method for dynamically controlling data traffic over a bi-directional data bus, said method comprising the step of allocating bus access between input direction traffic and output direction traffic based upon a relative demand for bus access between each of said traffic types.

2. The method of claim 1 wherein said network processor further comprises an aggregate ingress queue for buffering said input direction traffic and an aggregate egress queue for buffering said output direction traffic, and wherein the method further comprises determining said relative demand by determining an occupancy value for each of said aggregate queues and determining an input/output bias factor from said occupancy values, said input/output bias factor being representative of said relative demand.

3. The method of claim 2 wherein said allocating step further comprises dividing said bus access into a plurality of windows, each of said windows having an input scheduling frame and an output scheduling frame, wherein said input direction traffic is permitted access to said bus during said input scheduling frame, and wherein said output direction traffic is permitted access to said bus during said output scheduling frame.

4. The method of claim 3 wherein said allocating step further comprises the step of determining a duration for each of said input scheduling frames and said output scheduling frames in each of said windows.

5. The method of claim 4 wherein said step of determining said relative demand further comprises determining said relative demand in response to a bias damping factor, said bias damping factor at least partially defining a proportional relationship between said input/output bias factor, said input scheduling frame duration, and said output scheduling frame duration.

6. The method of claim 5 further comprising the step of selecting said bias damping factor.

7. The method of claim 5 wherein said step of determining either one of said input scheduling frame duration or said output scheduling frame duration comprises determining said duration in response to a target number of transactions therefor.

8. The method of claim 7 wherein said data traffic comprises a plurality of words, wherein said network processor further comprises a plurality of input queues for buffering words of said input direction traffic received from said aggregate ingress queue, and wherein said method further comprises the steps of reading a word buffered in said aggregate ingress queue, determining a relative occupancy for said input queues to identify an appropriate input queue for receiving said word, and writing said word into said appropriate input queue over said bus during said input scheduling frame.

9. The method of claim 8 further comprising the step of terminating the input scheduling frame prior to the end of its duration should the aggregate ingress queue become empty.

10. The method of claim 9 wherein the step of determining the relative occupancy comprises determining an occupancy for an input queue and comparing said occupancy with a threshold occupancy value.

11. The method of claim 8 further comprising the steps of determining from said word read from said aggregate ingress queue a desired destination input queue therefor and selecting said desired destination input queue as said appropriate input queue depending upon said comparison between said occupancy value for said desired destination input queue and said threshold occupancy value.

12. The method of claim 11 further comprising redirecting said word to another input queue if said desired destination input queue has an occupancy value exceeding said threshold occupancy value.

13. The method of claim 10 wherein the network processor further comprises a plurality of output queues for buffering words of said output direction traffic, and the method further comprises the steps of selecting an output queue, determining the occupancy for said selected output queue, assigning a number of transactions to said output queue at least in part according to said determined occupancy, reading a number of words from said output queue during said output scheduling frame up to said assigned number of transactions but no longer than said output scheduling frame duration, and repeating the foregoing steps until the end of said output scheduling frame duration.

14. The method of claim 12 further comprising the step of terminating the output scheduling frame prior to the end of its duration should the aggregate egress queue become full.

15. The method of claim 2 wherein each of said windows further has a pointer scheduling frame during which pointer traffic is permitted to access said bus.

16. The method of claim 15 further comprising the steps of recording at least two values corresponding to each pointer during each of said pointer scheduling frames, reading the recorded values corresponding to any one pointer during each of said pointer scheduling frames, and comparing the read values of said any one pointer for a match prior to using said any one pointer for writing or reading traffic.

17. A network processor for switching data between an input and an output, said network processor comprising a bi-directional data bus over which data traffic flows in an input direction and an output direction, said bus linking an aggregate ingress queue with a data processor in said input direction and linking said data processor with an aggregate egress queue in said output direction, and a queue manager connected in circuit with said aggregate queues and said bus, said queue manager being configured to dynamically allocate bus access between traffic in each direction based upon a relative demand for bus access between each of said traffic directions.

18. The network processor of claim 17 wherein said queue manager is further configured to (1) determine an input/output bias factor for said bus, (2) determine said relative demand for bus access as a function of said input/output bias factor, and (3) dynamically allocate bus access by dividing each of a plurality of windows of time on said bus into an input scheduling frame and an output scheduling frame, wherein said input direction traffic is permitted to access said bus during said input scheduling frame, wherein said output direction traffic is permitted to access said bus during said output scheduling frame, and wherein each input scheduling frame and each output scheduling frame has a duration that is dynamically allocated according to said determined relative demand.

19. The network processor of claim 18 wherein said queue manager is further configured to determine said relative demand as a function of a bias damping factor.

20. The network processor of claim 18 wherein said queue manager is further configured to (1) determine a target number of transactions for said input scheduling frame and said output scheduling frame, (2) assign a duration to said input scheduling frame sufficient to accommodate said determined target number of transactions therefor, and (3) assign a duration to said output scheduling frame sufficient to accommodate said determined number of transactions therefor.

21. The network processor of claim 18 wherein said queue manager is further configured to (1) determine an occupancy value for said aggregate ingress queue, (2) determine an occupancy value for said aggregate egress queue, and (3) determine said input/output bias factor as a function of said determined occupancy values for said aggregate ingress queue and said aggregate egress queue.

22. The network processor of claim 18 further comprising a plurality of input queues for buffering said input direction traffic and a plurality of output queues for buffering said output direction traffic, wherein said input queues and said output queues interface

said data processor with said bus, and wherein said data processor comprises a plurality of parallel processing elements, each parallel processing element being associated with at least one of said input queues and at least one of said output queues and being configured to (1) receive input direction traffic from said at least one associated input queue, (2) process said received input direction traffic, and (3) provide said processed traffic to said at least one associated output queue, wherein said processed traffic comprises said output direction traffic.

23. The network processor of claim 22 wherein said queue manager is further configured to (1) determine an occupancy value for each input queue, (2) during said input scheduling frame, select one of said input queues and write a word stored in said aggregate ingress queue to said selected input queue over said bi-directional data bus if said determined occupancy value for said selected input queue is less than a predetermined threshold occupancy value.

24. The network processor of claim 22 wherein said queue manager is further configured to, during said output scheduling frame, (1) select one of said output queues, (2) determine an occupancy value for said selected output queue, (3) determine a target number of output transactions for said selected output queue as a function of said determined occupancy value for said selected output queue, and (4) read a number of words from said selected output queue over said bi-directional data bus, wherein said determined number of output transactions for said selected output queue represents a maximum number of words read from said selected output queue during said output scheduling frame, and wherein the foregoing steps are repeated if said output scheduling frame is still active when said queue manager is finished reading words from said selected output queue.

25. The network processor of claim 24 wherein said plurality of input queues and said plurality of output queues are implemented on a plurality of dual port RAMs (DPRAMs), each DPRAM comprising at least one of said input queues and at least one of said output queues.

26. The network processor of claim 25 wherein said plurality of input queues and output queues are implemented on a plurality of DPRAMs, each DPRAM comprising two input queues and two output queues.

27. The network processor of claim 24 wherein said queue manager and said data processor are in mutually asynchronous clock domains.

28. The network processor of claim 22 wherein said queue manager is further configured to include a pointer scheduling frame in each of said windows, wherein pointer traffic is permitted to access said bus during each pointer scheduling frame.

29. The network processor of claim 24 wherein each DPRAM further comprises a plurality of stored pointer values for each input queue and each output queue implemented on said DPRAM, said stored pointer values comprising two copies of an input write pointer for each input queue implemented on said DPRAM, two copies of an input read pointer for each input queue implemented on said DPRAM, two copies of an output write pointer for each output queue implemented on said DPRAM, two copies of an output read pointer for each output queue implemented on said DPRAM, wherein said queue manager is further configured to (1) read both copies of an input read pointer and both copies of an output write pointer stored in a DPRAM during each pointer scheduling frame, (2) write both copies of an input write pointer and both copies of an output read pointer stored in a DPRAM during each pointer scheduling frame, wherein said data processor is configured to (1) write both copies of an input read pointer and both copies of an output write pointer stored in a DPRAM, (2) read both copies of an input write pointer and both copies of an output read pointer stored in a DPRAM, and wherein said queue manager is further configured to read and write said pointer copies in a different order than said data processor is configured to do so.

30. The network processor of claim 29 wherein said queue manager is further configured to (1) compare both copies of a read input read pointer to determine whether a match exists, and (2) update an internal input read pointer value with said read input read pointer only if said comparison results in a match, (3) compare both copies of a read output write pointer to determine whether a match exists,

and (2) update an internal output write pointer value with said read output write pointer only if said comparison results in a match.

31. The network processor of claim 24 further comprising a plurality of additional bi-directional data buses over which said input direction traffic and said output direction traffic flows, each additional bi-directional data bus linking one of said parallel processing elements with its at least one associated input queue in an input direction and with its at least one associated output queue in an output direction, and wherein each parallel processing element is configured to dynamically allocate access to said additional bus to which it is linked between traffic in each direction based upon a relative demand for access to said additional bus between each of said traffic directions.

32. A device for dynamically controlling data traffic over a bi-directional data bus, said device comprising a data scheduler configured to allocate bus access between input direction traffic and output direction traffic based upon a relative demand for bus access between each of said traffic types.

33. The device of claim 32 wherein said device is linked to an aggregate ingress queue from which it dequeues input direction traffic and linked to an aggregate egress queue to which it queues output direction traffic, and wherein said data scheduler is further configured to determine said relative demand by determining an occupancy value for each of said aggregate queues and determining an input/output bias factor from said occupancy values, said input/output bias factor being representative of said relative demand.

34. The device of claim 33 further comprising a main scheduler configured to divide said bus access into a plurality of windows, each of said windows having a data scheduling frame, and wherein said data scheduler is further configured to divide each of said data scheduling frames into an input scheduling frame and an output scheduling frame, wherein said input direction traffic is permitted access to said bus during said input scheduling frame, and wherein

said output direction traffic is permitted access to said bus during said output scheduling frame.

35. The device of claim 34 wherein said data scheduler is further configured to allocate said bus access by determining a duration for each of said input scheduling frames and said output scheduling frames in each of said windows.

36. The device of claim 35 wherein said data scheduler is further configured to determine said relative in response to a bias damping factor, said bias damping factor at least partially defining a proportional relationship between said input/output bias factor, said input scheduling frame duration, and said output scheduling frame duration.

37. The device of claim 36 wherein said bias damping factor is predetermined.

38. The device of claim 36 wherein said data scheduler is further configured to (1) determine a target number of transactions for either one of said input scheduling frame or said output scheduling frame, and (2) determine either one of said input scheduling frame duration or said output scheduling frame duration in response to said determined target number of transactions.

39. The device of claim 38 wherein said data traffic comprises a plurality of words, wherein said device is further linked via said busto a plurality of input queues for buffering words of said input direction traffic received from said aggregate ingress queue, and wherein said device further comprises an input scheduler configured to (1) read a word buffered in said aggregate ingress queue, (2) determine a relative occupancy for said input queues to identify an appropriate input queue for receiving said word, and (3) write said word into said appropriate input queue over said bus during said input scheduling frame.

40. The device of claim 39 wherein said input scheduler is further configured to terminate an input scheduling frame prior to the end of its duration should the aggregate ingress queue become empty.

41. The device of claim 40 wherein said input scheduler is further configured to determine the relative occupancy for an input queue by determining an occupancy for that input queue and comparing said occupancy with a threshold occupancy value.

42. The device of claim 41 wherein said input scheduler is further configured to determine from said word read from said aggregate ingress queue a desired destination input queue therefor, and select said desired destination input queue as said appropriate input queue depending upon said comparison between said occupancy value for said desired destination input queue and said threshold occupancy value.

43. The device of claim 41 wherein said device is further linked via said bus to a plurality of output queues for buffering words of said output direction traffic, and wherein said device further comprises an output scheduler configured to (1) select an output queue, (2) determine the occupancy for said selected output queue, (3) assign a number of transactions to said output queue at least in part according to said determined occupancy, (4) read a number of words from said output queue during said output scheduling frame up to said assigned number of transactions but no longer than said output scheduling frame duration, and (5) repeat the foregoing steps until the end of said output scheduling frame duration.

44. The device of claim 43 wherein said output scheduler is further configured to terminate the output scheduling frame prior to the end of its duration should the aggregate egress queue become full.

45. The device of claim 44 wherein each of said windows further includes a pointer scheduling frame during which pointer traffic is permitted to access said bus.

46. The device of claim 45 further comprising a pointer scheduler configured to (1) record during each pointer scheduling frame at least two values corresponding to a pointer, (2) read during each pointer scheduling frame the recorded values corresponding to any one pointer, and (3) compare the read values of said any one pointer for

a match prior to using said any one pointer for writing or reading traffic.

47. A method of dynamically controlling data flow on a bi-directional data bus, wherein said data flow comprises input direction traffic traveling over said bi-directional data bus from an aggregate ingress queue (AIQ) to a plurality of input queues (IQs) and output direction traffic traveling over said bi-directional data bus from a plurality of output queues (OQs) to an aggregate egress queue (AEQ), wherein said input direction traffic and said output direction traffic are both comprised of a plurality of words, and wherein each of said queues has a plurality of addresses therein for storing words, said method comprising:

a) dividing a window into a data scheduling frame having a size sufficient for a number of word transactions and a pointer scheduling frame;

b) determining an input/output bias factor for said bi-directional data bus;

c) dividing said data scheduling frame into an input scheduling frame and an output scheduling frame such that said input scheduling frame and said output scheduling frame have sizes relative to each other that depend at least in part upon said determined input/output bias factor;

d) during said input scheduling frame, writing a number of words stored in said AIQ to at least one of said IQs over said bi-directional data bus;

e) during said output scheduling frame, reading a number of words stored in at least one of said OQs and destined for said AEQ over said bi-directional data bus; and

f) during said pointer scheduling frame, updating at least one pointer for at least one of said IQs or at least one of said OQs, wherein each pointer identifies a queue address to which a next word will be written or from which a next word will be read.

48. The method of claim 47 wherein said input/output bias factor determining step includes:

determining a value representing an occupancy of said AIQ;

determining a value representing an occupancy of said AEQ; and

calculating said input/output bias factor as a function of said determined AIQ occupancy value and said determined AEQ occupancy value.

49. The method of claim 48 wherein said data scheduling frame dividing step includes:

calculating a number of input transactions for said input scheduling frame as a function of said input/output bias factor and said number of word transactions;

assigning a duration to said input scheduling frame sufficient to accommodate said determined number of input transactions; and

assigning a duration to said output scheduling frame equal to a remainder of said data scheduling frame.

50. The method of claim 49 wherein said step of calculating said number of input transactions for said input scheduling frame includes calculating said number of input transactions as a function of a predetermined bias damping coefficient.

51. The method of claim 50 wherein said input/output bias factor determining step further includes:

determining a word capacity for said AIQ;

determining a word capacity for said AEQ; and

wherein said input/output bias factor calculation step includes calculating said input/output bias factor according to the formula: input/output bias factor = (said determined occupancy of said AIQ + said determined occupancy of said AEQ) - ((said determined word capacity of said AIQ + said determined word capacity of said AEQ)/2).

52. The method of claim 51 wherein said step of calculating said number of input transactions for said input scheduling frame further includes calculating said number of input transactions for said input scheduling frame according to the formula: said number of input transactions for said input scheduling frame = (said number of word transactions for said data scheduling frame/2) + (said determined input/output bias factor * said predetermined bias damping factor)

53. A method of dynamically controlling data flow on a bi-directional data bus, wherein said data flow comprises input

direction traffic traveling over said bi-directional data bus from an aggregate ingress queue (AIQ) to a plurality of input queues (IQs) and output direction traffic traveling over said bi-directional data bus from a plurality of output queues (OQs) to an aggregate egress queue (AEQ), wherein said input direction traffic and said output direction traffic are both comprised of a plurality of words, and wherein each of said queues has a plurality of addresses therein for storing words, said method comprising:

a) dividing a window into a data scheduling frame having a size sufficient for a number of word transactions and a pointer scheduling frame;

b) dividing said data scheduling frame into an input scheduling frame and an output scheduling frame;

c) for each IQ, determining a value representing an occupancy thereof;

d) during said input scheduling frame, writing a number of words stored in said AIQ to at least one of said IQs over said bi-directional data bus by, for each of said number of words, (1) selecting an IQ in which to write said word stored in said AIQ according to said determined occupancy value for that IQ relative to a predetermined threshold value, and (2) writing said word stored in said AIQ to an address in said selected IQ over said bi-directional data bus;

e) during said output scheduling frame, reading a number of words stored in at least one of said OQs and destined for said AEQ over said bi-directional data bus; and

f) during said pointer scheduling frame, updating at least one pointer for at least one of said IQs or at least one of said OQs, wherein each pointer identifies a queue address to which a next word will be written or from which a next word will be read.

54. The method of claim 53 wherein said IQ selecting step includes:

determining a desired destination IQ for said word stored in said AIQ;

if said determined occupancy value for said determined IQ is less than a predetermined threshold occupancy value, selecting said determined IQ;

if said determined occupancy value for said determined IQ is greater than or equal to said predetermined threshold occupancy value

and if at least one other of said IQs has a determined occupancy less than said predetermined threshold occupancy value, selecting one of said IQs having a determined occupancy less than said predetermined threshold occupancy value; and

if all of said IQs have a determined occupancy value greater than or equal to said predetermined threshold occupancy value, terminating said input scheduling frame.

55. A method of dynamically controlling data flow on a bi-directional data bus, wherein said data flow comprises input direction traffic traveling over said bi-directional data bus from an aggregate ingress queue (AIQ) to a plurality of input queues (IQs) and output direction traffic traveling over said bi-directional data bus from a plurality of output queues (OQs) to an aggregate egress queue (AEQ), wherein said input direction traffic and said output direction traffic are both comprised of a plurality of words, and wherein each of said queues has a plurality of addresses therein for storing words, said method comprising:

a) dividing a window into a data scheduling frame having a size sufficient for a number of word transactions and a pointer scheduling frame;

b) dividing said data scheduling frame into an input scheduling frame and an output scheduling frame;

c) during said input scheduling frame, writing a number of words stored in said AIQ to at least one of said IQs over said bi-directional data bus;

d) during said output scheduling frame, reading a number of words destined for said AEQ and stored in at least one of said OQs over said bi-directional data bus by, for each of said number of stored words destined for said AEQ, (1) selecting a non-empty OQ, (2) determining a value representing an occupancy for said selected OQ, (3) determining a number of output transactions for said selected OQ as a function of said determined occupancy value for said OQ, (4) if said output scheduling frame has a duration sufficient to accommodate said determined number of output transactions, reading a number of words from said selected OQ equal to said determined number of output transactions, and (5) if said output scheduling frame duration is insufficient to accommodate said determined number of output

transactions, reading words from said selected IQ until said output scheduling frame terminates; and

e) during said pointer scheduling frame, updating at least one pointer for at least one of said IQs or at least one of said OQs, wherein each pointer identifies a queue address to which a next word will be written or from which a next word will be read.

56. The method of claim 55 wherein said step of determining a number of output transactions includes determining said number of output transactions as a function of a predetermined occupancy damping factor.

57. The method of claim 56 wherein said step of determining a number of output transactions further includes determining said number of output transactions as a function of a predetermined baseline number of output transactions.

58. The method of claim 55 wherein said step of determining a number of output transactions includes determining said number of output transactions as a function of a predetermined baseline number of output transactions.

59. The method of claim 55 wherein said OQ selecting step includes selecting a non-empty OQ in a round-robin fashion.

60. A method of controlling data flow on a bi-directional data bus, wherein said data flow comprises input direction traffic traveling over said bi-directional data bus from an aggregate ingress queue (AIQ) to a plurality of input queues (IQs) and output direction traffic traveling over said bi-directional data bus from a plurality of output queues (OQs) to an aggregate egress queue (AEQ), wherein said input direction traffic and said output direction traffic are both comprised of a plurality of words, and wherein each of said queues has a plurality of addresses therein for storing words and pointers, wherein each IQ has stored therein at least two copies of an input read pointer and at least two copies of an input write pointer, wherein each OQ has stored therein at least two copies of an output read pointer and at least two copies of an input read pointer, said method comprising:

a) dividing a window into a data scheduling frame having a duration sufficient for a number of word transactions and a pointer scheduling frame;

b) dividing said data scheduling frame into an input scheduling frame and an output scheduling frame;

c) during said input scheduling frame, writing a number of words stored in said AIQ to at least one of said IQs over said bi-directional data bus;

d) during said output scheduling frame, reading a number of words stored in at least one of said OQs and destined for said AEQ over said bi-directional data bus; and

e) during said pointer scheduling frame, updating said pointers for at least one of said IQs and at least one of said OQs, by (1) selecting an IQ and an OQ, (2) reading both copies of the input read pointer stored in said selected IQ, (3) reading both copies of the output write pointer stored in said selected OQ, (4) comparing both copies of said input read pointers to determine whether a match exits, (5) comparing both copies of said output write pointers to determine whether a match exists, (6) if a match exists between said input read pointer copies, updating an internal input read pointer value for said selected IQ with said read input read pointer, and (7) if a match exists between said output write pointer copies, updating an internal output write pointer value for said selected OQ with said read output write pointer.

61. The method of claim 60 wherein said pointer updating step includes:

in said selected IQ, overwriting both copies of said stored input write pointer with an internal input write pointer; and

in said selected OQ, overwriting both copies of said stored output read pointer with an internal output read pointer.

62. The method of claim 60 wherein said step of selecting an IQ and an OQ includes selecting an IQ and an OQ in a round-robin fashion.